**GOTO 2016 • What I Wish I Had Known Before Scaling Uber to 1000 Services • Matt Ranney**

**Sobre el Autor**

Actual ingeniero Senior de Uber en la cual labora 4 años ayudando a diseñar e implementar sistemas backend usando JavaScript (node.js), Python, Go, y Java.

**UBER**

El autor comenta el panorama actual de Uber que para el 2006 se encuentra presente en más de 400 ciudades, 70 países y a llegado a tener más de 6000 empleados, debido a que su crecimiento se ha multiplicado por 10 en el último año y medio.

**Lecciones de Vida**

El aprendizaje que ha tenido en este corto tiempo en Uber le ha permitido ha sido muy grande y necesario para poder liderar y protagonizar sus logros.

**Servicios**

El desarrollo exponencial de servicios en Uber llegando a desarrollar 106 servicios por semana como punto máximo.

**MicroServicios**

Hay gran dificultad que genera el desarrollo de una arquitectura de micro servicios debido a la gran cantidad de grupos de trabajo y lenguajes que se emplean en todo el proceso.

Se sabe que los sistemas funcionan mejor cuando no están en constante cambio, pero estos cambios son importantes y deben realizarse para mejorar el funcionamiento del sistema en general.

**¿Por qué usar Microservicios?**

la capacidad de mover independientemente las partes del sistema, poder organizar adecuadamente los tiempos y poder implementar diferentes herramientas según sea conveniente dentro del mismo sistema es la respuesta adecuada a esta pregunta

**¿Cuáles son los costos?**

Los sistemas distribuidos pueden generar inconsistencias que serán difíciles de corregir más adelante, todo se encuentra en un RPC, lo que puede generar que si el sistema se cayera, toda la red dejaría de funcionar adecuadamente y habría una pérdida mayor del servicio por ese plazo de tiempo.

**Costos menos obvios**

Como la existencia de una compensación para todo, si bien puedes arreglar un problema, siempre habrá una reacción, además se puede construir sobre errores buscando taparlos, generando un error permanente en el sistema; la complejidad de las políticas del negocio también se pueden ver influenciadas y se generan prejuicios muy difíciles de romperdebido a las preferencias de los trabajadores y los grupos que se forman naturalmente dentro de la cultura de la organización.

**Lenguajes**

La posibilidad de usar diversos lenguajes de programación según sea necesario puede ser un beneficio para el desarrollo, también es una desventaja para la administración de todos estos equipos de desarrollo. Los costos que se generarán incluyen la dificultad para compartir código y la fragmentación de la cultura, como se mencionó anteriormente, pues se forman opiniones y grupos relativos al lenguaje y frameworks que se emplean por cada servicio.

**RPC**

Se generan varias complicaciones relativas a http/rest, la necesidad de generar un esquema general para json, la lentitud frente a pcs y la confusión de un servidor con un buscador.

**Repositorios**

El autor habla de la cantidad de repositorios necesarios y se menciona que uno sería una buena opción, pero en la práctica Uber posee más de 4000 repositorios y sigue en aumento directamente proporcional al aumento de personal de la compañia.

**Operaciones**

la existencia de grupos y diversos métodos de trabajo dificulta la ayuda de otros grupos, pues difícil ver el sistema como un todo, cuando se encuentra dividido en tantas partes más pequeñas. Surge el que pasaría si las cosas dejan de funcionar? , otros equipos podrían asumir las funciones?,.

**Performance**

Se depende de las herramientas del lenguaje que son difíciles de medir cuando estas son tan diferentes, por lo cual se buscó una solución más estándar, proponiendo un dashboard general que todas las áreas debían utilizar para obtener una comparación adecuada del desempeño a nivel de toda la empresa entre ellos el grafico Flama.